This means that the branching mechanism described above can be used to switch back from the second instruction set to the first instruction set.

An example of an instruction set selection sequence (known as a "veneer") is as follows:

Label_Veneer	
XOR	(PC,1)
Branch	Label

In this routine, the current contents of the program counter register 130" of FIG. 3 are exclusive-ORed with 1 to set the T bit to 1. (Alternatively, with the program counter 130' of FIG. 2, the current contents could be exclusive-ORed with 10000000000000000000000000000000 to set the T bit).

In an alternative veneer routine, a subtract operation could be used instead of an exclusive-OR operation to change the T-bit of the program counter register 130". This has the advantage that in some processors, the subtract operation also flushes or clears the instruction pipeline 80.

The following example assumes that the program counter 130" points 8 bytes beyond the current instruction, and that the current instruction is a 32 bit (4 byte) instruction. Accordingly, to change the least significant bit of the program counter register 130" to 1, it is necessary to add or subtract the following amounts to the current program counter register contents:

add 1	(to change the T bit to 1)
subtract 8	(to compensate for the program counter pointing ahead of the current instruction)
add 4	(to compensate for the length of the current instruction)
----------	
subtract 3	(total change)

The instruction sequence used is therefore:

Label_Veneer		
SUB	(PC,PC,3)	(replace PC with PC-3)
Branch	Label	

In summary, the use of the program counter to store the instruction-set-specifying bit or bits has at least the following advantages:

1. It provides a single, uniform method of identifying a target routine by representing both the target address and the corresponding instruction set in a single machine word.

2. The code size is reduced as fewer veneers are required.

3. The processor performance can be improved as there is no longer a need to execute a veneer on each inter-instruction set routine call.

Although illustrative embodiments of the invention have been described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various changes and modifications can be effected therein by one skilled in the art without departing from the scope and spirit of the invention as defined by the appended claims.

I claim:

1. Data processing apparatus comprising:

(i) a processor core operable to execute successive program instruction words of a predetermined plurality of instruction sets stored in a data memory;

(ii) a program counter register for indicating an address of a next program instruction word in said data memory;

(iii) logic operable to modify the contents of said program counter register in response to a current program instruction word;

(iv) a processor core controller, responsive to one or more predetermined indicator bits of said program counter register, operable to control said processor core to execute program instruction words of a current instruction set selected from said predetermined plurality of instruction sets and specified by the state of said one or more indicator bits of said program counter register; and

(v) a memory access controller operable to access program instruction words stored in said data memory, said access controller not being responsive to said one or more indicator bits of said program counter register.

2. Apparatus according to claim 1, comprising:

a first instruction decoder for decoding program instruction words of a first instruction set; and

a second instruction decoder for decoding program instruction words of a second instruction set;

and in which said processor core controller is operable to control either said first instruction decoder or said second instruction decoder to decode a current program instruction word.

3. Apparatus according to claim 2, in which:

program instruction words of said first instruction set are X-bit program instruction words; and

program instruction words of said second instruction set are Y-bit program instruction words;

Y being different to X.

4. Apparatus according to claim 1, in which:

program instruction words of a first instruction set are X-bit program instruction words; and

program instruction words of a second instruction set are Y-bit program instruction words;

Y being different to X.

5. Apparatus according to claim 3, in which Y is 16 and X is 32.

6. Apparatus according to claim 4, in which Y is 16 and X is 32.

7. Apparatus according to claim 1, in which said one or more indicator bits of said program counter register are one or more most significant bits of said program counter register.

8. Apparatus according to claim 1, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

9. Apparatus according to claim 2, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

10. Apparatus according to claim 3, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

11. Apparatus according to claim 4, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

12. Apparatus according to claim 5, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

13. Apparatus according to claim 6, in which said one or more indicator bits of said program counter register are one or more least significant bits of said program counter register.

14. Apparatus according to claim 1, comprising a data memory for storing program instruction words to be executed.

* * * * *